

What the dispatcher does

“ **Audience:** Operations and dispatch managers who configure, supervise, or explain automatic order assignment to their team. **Purpose:** Understand *what* the dispatcher does and *why* it makes a given decision.

The Dispatcher is the system that automatically decides **which driver (or external carrier) delivers which order**, and **when**. It runs on its own, on a schedule you control, and re-evaluates every store's pending orders each time it runs.

These pages explain the logic behind those decisions in business terms: what the algorithm optimises for, what determines priority, which settings are available to you, and what to expect — or not expect — from it.

How to read this chapter

You don't need to read it front to back. Most questions land on one page:

If you want to know...	Go to
What a term means	Dispatcher glossary
What happens on each run	The assignment cycle, step by step
Why an order stayed with an external carrier	Re-checking orders already with a carrier
Why one driver was chosen over another	Driver classification and priority
Why an order changed driver	When an order changes driver
Why an order was left unassigned	Last-Option: the final safety net and the Dispatcher FAQ
How often it runs, and how to start or stop it	Scheduling plans: how the automatic cycle works
Where to change a particular setting	Where each setting lives
The whole flow on one page	End-to-end overview
A rule, without the explanation	Quick reference rules
How to report something that looks wrong	Reporting a dispatcher issue

Once the dispatcher has made its decisions, the live board where a person can intervene is the Hub — see [The Hub: what it is and who can do what](#).

What it does not do

The dispatcher is deliberately bounded. It does not try to find a mathematically perfect plan across every driver and every order at once, it never touches an order that has already left the store, and it does not reorder work just because one order is more urgent than another. Those limits are listed in full under [Known system boundaries](#) — they are design decisions, not faults, and knowing them saves a lot of guesswork.

Created 2026-09-17 16:28:17 UTC by Fran Sanchis

Updated 2026-09-18 07:48:12 UTC by Fran Sanchis